

日本国特許庁  
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2000年 9月20日

出願番号

Application Number:

特願2000-286066

出願人

Applicant(s):

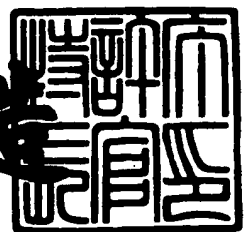
株式会社東芝

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 4月27日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3036785

【書類名】 特許願

【整理番号】 A000004450

【提出日】 平成12年 9月20日

【あて先】 特許庁長官 殿

【国際特許分類】 H04N 7/30

【発明の名称】 通信方法および情報処理装置

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

【氏名】 木村 哲郎

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 通信方法および情報処理装置

【特許請求の範囲】

【請求項 1】 プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも 1 つ有する複数の端末があり、これらのうちの特定の端末が、自装置のネットワークアドレスの変更を検知したとき、他の端末が前記特定の端末へのアクセスを可能にするため、変更後の新たなネットワークアドレスを前記他の端末が参照可能なように提供することを特徴とする通信方法。

【請求項 2】 前記特定の端末が、自装置のネットワークアドレスの変更を検知したとき、前記特定の端末に記憶されているファイルであって前記特定の端末が前記他の端末に提供すべきファイルの所在を示す表記データを更新し、この更新された表記データに基づいて前記ファイルを前記他の端末が参照可能なように提供することを特徴とする請求項 1 記載の通信方法。

【請求項 3】 プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも 1 つ有し、前記実行環境上にて、クライアントからの要求に応じて所定のサービスを提供するためのサーバプログラムを実行する情報処理装置であって、

自装置のネットワークアドレスの変更を検知する検知手段と、

この検知手段でネットワークアドレスの変更を検知したとき、前記クライアントが自装置へのアクセスを可能にするため、少なくとも変更後の新たなネットワークアドレスを前記クライアントが参照可能なように提供する提供手段と、

を具備したことを特徴とする情報処理装置。

【請求項 4】 前記クライアントが前記サービスの提供を受けるために必要な手続きを記述したファイルを記憶する記憶手段と、

前記検知手段でネットワークアドレスの変更を検知したとき、前記ファイルの所在を示した表記データを更新する更新手段と、

更新された表記データに基づいて前記ファイルを前記クライアントが参照可能なように提供する提供手段と、

を具備したことを特徴とする請求項 3 記載の情報処理装置。

【請求項 5】 プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも 1 つ有するコンピュータを、クライアントからの要求に応じて所定のサービスを提供するための前記実行環境上のサーバとして機能させるためのプログラム製品であって、

自装置のネットワークアドレスの変更を検知する機能と、

前記ネットワークアドレスの変更を検知したとき、前記クライアントが自装置へのアクセスを可能にするため、少なくとも変更後の新たなネットワークアドレスを前記クライアントが参照可能なように提供する機能と、

前記ネットワークアドレスの変更を検知したとき、前記クライアントが前記サービスの提供を受けるために必要な手続きを記述したファイルの記憶場所を示した表記データを更新する機能と、

更新された表記データに基づいて前記ファイルを前記クライアントが参照可能なように提供する機能と、

をコンピュータに実現させるためのプログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は J a v a (サン・マイクロシステムズ登録商標) 実行環境による分散情報処理システムに関する。

【0002】

【従来の技術】

近年、プラットフォーム独立なプログラム実行環境 J a v a (米サンマイクロシステムズ社登録商標、文献 Ken Arnold, James Gosling, " The Java Programming Language, Second Edition, " 参照) の発展により、多種多様な情報機器を J a v a により連携させる取り組みが盛んに行われている。

【0003】

分散アプリケーションを J a v a で実装する場合、通信手段として遠隔メソッ

ド呼び出し(Remote Method Invocation 以下RMIと呼ぶ)という通信技術を利用するのが一般的である(文献 Troy Bryan Downing、"RMI:Developing Distributed Java Applications with Remote Method Invocation and Serialization," IDG Books WorldWide, Inc. 1998参照)。

【0004】

RMIは、Javaのオブジェクト間のメソッド呼び出し機構を通信用に拡張したものであり、複雑な通信処理を隠蔽した、簡単でかつ強力な通信機構である。

【0005】

【発明が解決しようとする課題】

しかし、JavaおよびRMIは、実行される計算機(ホスト)のIPアドレス(以下、ネットワークアドレスとも呼ぶ)の変更に対応できないという問題がある。具体的には、以下の2点がある。

【0006】

(1) RMIでは、クライアントがサーバに接続する際、リモートリファレンス(リモート参照)と呼ばれる情報を利用して接続する。このリモート参照には、サーバプログラムが実行されている計算機のネットワークアドレス(IPアドレス)とポート番号などの情報が含まれている。この情報(リモート参照)は、例えば、計算機で電源が投入されて(起動されて)、サービスを公開する時にサーバプログラムが作成するが、作成する時のネットワークアドレスがリモート参照に埋め込まれてしまい、その後のネットワークアドレスの変更に対応して更新するメカニズムを持たない。

【0007】

そのため、計算機が移動して、ネットワークアドレス(以下、簡単にネットワークとも呼ぶ)が変更された後に、サーバプログラムが配布する(ディレクトリサービスプログラムに登録する)リモート参照には、移動前のネットワークアドレスが埋め込まれている。このリモート参照を受け取ったクライアントプログラ

ムは、移動前のネットワークアドレスに対してコネクションを張ろうと試みてしまい、新しいネットワークアドレスを持ったサーバとのコネクションが確立できないという問題点があった。

【0008】

(2) RMIではサーバとクライアント間でJ a v aオブジェクトを送受信する。受信側のJ a v aプログラムは、受け取ったJ a v aオブジェクトに対応するクラスファイルを必要とする。このクラスファイルを既に持っている場合には、それを利用して受信したJ a v aオブジェクトの処理を行う。しかし、このクラスを持っていない場合には、J a v aオブジェクトに付与されているクラスファイルの供給元の情報を参照し、そこからダウンロードを行い、その後にJ a v aオブジェクトの処理を行う。J a v aオブジェクトに付与されたクラスファイルの供給元情報はコードベースと呼ばれ、U n i f o r m R e s o u r c e L o c a t o r (URL)形式で指定する。

【0009】

このコードベースも前述のリモート参照と同様に設定は静的であり、ネットワークアドレスの変更に対応して更新するメカニズムを持たない。このため、ネットワーク移動を行う計算機のように、動的にネットワークアドレスが変更される計算機がクラスファイル供給元となる場合に、コードベースの更新が行われず、受信側のプログラムが正常に動作できないという問題点があった。

【0010】

そこで、本発明は、上記問題点に鑑み、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも1つ有する端末のI Pアドレスが、例えば上記プログラム実行中に不意に変更されても、そのI Pアドレスの変更を上記実行環境上に反映して、後に、再び他の端末との通信を可能にする通信方法および情報処理装置を提供することを目的とする。

【0011】

例えば、J a v a仮想マシンを少なくとも1つ有する端末のI Pアドレスの変更に伴い、RMIにて定義されているリモート参照中のI Pアドレス、コードベ

ースも容易に変更でき、IPアドレス変更後もJavaプログラムを用いた端末間の通信を可能にする通信方法およびそれを用いた情報処理装置を提供することを目的とする。

#### 【0012】

##### 【課題を解決するための手段】

本発明は、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境（例えばJava仮想マシン）を少なくとも1つ有する複数の端末があり、これらのうちの特定の端末（例えばサーバプログラムが実行される端末）が、自装置のネットワークアドレス（例えばIPアドレス）の変更を検知したとき、他の端末（例えば、クライアントプログラムが実行される端末）が前記特定の端末へのアクセスを可能にするため、変更後の新たなネットワークアドレスを前記他の端末が参照可能なように提供し（例えば、リモート参照中のネットワークアドレスを更新して、ディレクトリサービスプログラムへ再度登録し直す）、また、前記特定の端末が、自装置のネットワークアドレスの変更を検知したとき、前記特定の端末に記憶されているファイルであって、前記特定の端末が前記他の端末に提供すべきファイル（例えば、スタブファイルやその他のクラスファイル）の所在を示す表記データ（コードベース）を更新し、この更新された表記データに基づき前記ファイルを前記他の端末が参照可能なように提供する（例えば、スタブファイル中のコードベースを更新したり、必要に応じて、相手端末へ送付する）することにより、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境（例えば、Java仮想マシン）を少なくとも1つ有する端末のネットワークアドレスの変更に伴い、例えば、クライアントに提供されるリモート参照中のIPアドレス、コードベースも容易に変更でき、端末のIPアドレスが不意に変更された場合も（例えば、Javaプログラムを用いた）端末間の通信を可能にする。

#### 【0013】

特に、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたサーバプログラムを実行する実行環境（例えば、Java仮想マシン）

を少なくとも1つ有する端末が、自装置のネットワークアドレスの変更を検知する検知手段と、この検知手段でネットワークアドレスの変更を検知したとき、前記クライアントが自装置へのアクセスを可能にするため、少なくとも変更後の新たなネットワークアドレスを前記クライアントが参照可能なように提供する提供手段とを具備し、好ましくは、前記クライアントが前記サービスの提供を受けるために必要な手続きを記述したファイルを記憶する記憶手段と、前記検知手段でネットワークアドレスの変更を検知したとき、前記ファイルの所在を示した表記データ（例えば、コードベース）を更新する更新手段と、更新された表記データに基づき前記ファイルを前記クライアントが参照可能なように提供する提供手段とを具備したことにより、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたサーバプログラムを実行する実行環境（例えば、J a v a 仮想マシン）を少なくとも1つ有する端末のネットワークアドレスの変更に伴い、クライアントに提供するためにディレクトリサービスプログラムに登録するサーバプログラムへのリモート参照中のI Pアドレス、コードベースも容易に変更でき、サーバプログラム実行中にその端末のI Pアドレスが不意に変更されても、後にクライアントからのアクセス（例えば、遠隔メソッド呼び出し）を可能にし、（例えば、J a v a プログラムを用いた）端末間の通信を可能にする。

## 【 0 0 1 4 】

本発明は、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも1つ有するコンピュータを、クライアントからの要求に応じて所定のサービスを提供するための前記実行環境上のサーバとして機能させるためのプログラム製品であって、

自装置のネットワークアドレスの変更を検知する機能と、

前記ネットワークアドレスの変更を検知したとき、前記クライアントが自装置へのアクセスを可能にするため、少なくとも変更後の新たなネットワークアドレスを前記クライアントが参照可能なように提供する機能と、

前記ネットワークアドレスの変更を検知したとき、前記クライアントが前記サービスの提供を受けるために必要な手続きを記述したファイルの記憶場所を示した表記データを更新する機能と、

更新された表記データを前記クライアントが参照可能なように提供する機能と

をコンピュータに実現させる。本発明によれば、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境（例えば、J a v a 仮想マシン）を少なくとも1つ有するコンピュータのネットワークアドレス（例えば、I P アドレス）の変更に伴い、クライアントに提供されるリモート参照中のI P アドレス、コードベースも容易に変更でき、移動先においても（例えば、J a v a プログラムを用いた）コンピュータ間の通信を可能にする。

【0015】

#### 【発明の実施形態】

図1は、本実施形態に係る分散情報処理システムの構成例を示す図で、プラットフォーム独立の機械語で実行されるオブジェクト指向言語として、J a v a で記述されたプログラムを実行する実行環境（J a v a 仮想マシン）を少なくとも1つ有する複数（例えば、ここでは3つの）計算機101～103が、例えばインターネット等のネットワークを介して接続され、J a v a のR M I を用いてJ a v a プログラム間で通信を行うことにより、分散情報処理を実現するものである。

【0016】

J a v a のR M I を用いて分散情報処理を実現するためには、サービスを提供するサーバプログラムと、サービスを利用するクライアントプログラム、およびサーバとクライアントを仲介するディレクトリサービスプログラムが必要である。

【0017】

図1において、サーバプログラム、クライアントプログラム、ディレクトリサービスプログラムは、それぞれ、異なる計算機（計算機102、計算機103、計算機101）上のJ a v a 仮想マシン（V i r t u a l M a c h i n e）S、C、Dで実行されているプログラムである（同じ計算機上の異なるJ a v a 仮想マシンであっても一般性を失わない）。

## 【0018】

ディレクトリサービスプログラムは、クライアントプログラムがサーバプログラムを発見するための手段を提供するプログラムで、クライアントプログラムから要求されたサービス名やサービス属性などをキーとして、サーバプログラムを検索する機能を有する。例えば、Javaの開発環境に含まれるrmiregistryというプログラムや、Sun Microsystems社が提唱するJini(文献 K. Arnold, B. O' Sullivan, R. W. Scheifler, J. Waldo, and A. Wollrath, "The Jini Specification," Addison-Wesley, 1999 参照)のLookup Serverというプログラムがこれにあたる。

## 【0019】

サーバプログラムは、他のJava仮想マシン上のクライアントプログラムからのサービス要求を遠隔メソッド呼び出しの形で受信し、サービス結果をメソッドへの返回值として返す。サーバプログラムは、サービス提供を開始するにあたり、まずディレクトリサービスプログラムに、そのサーバプログラムから提供するサービス名や属性と、当該サーバプログラムへの接続に必要な情報(ネットワークアドレス(ネットワークレイヤアドレスで、ここでは、IPアドレス)、ポート番号、オブジェクト識別子などで、これを後述するスタブファイルに埋め込んで登録する。

## 【0020】

クライアントプログラムは、サーバプログラムに対して遠隔メソッド呼び出しを用いて要求を発行し、応答を受け取って目的を完遂する。クライアントプログラムは、サービスを受けるにあたってまずサーバプログラムが稼働するネットワーク上の位置を知らなければならない。そこで、クライアントプログラムはまずディレクトリサービスプログラムに接続して、サービス名やサービス属性などをキーとして、所望のサーバプログラムの検索を行なう。検索に成功すると、ディレクトリサービスプログラムに登録されているリモート参照を獲得するので、リモート参照に基づきサーバプログラムへ接続し、サービスの提供を受ける。

【0021】

J a v a 仮想マシンは、R M I を利用するために、同じ J a v a 仮想マシン内にリモート参照管理部を具備する。

【0022】

リモート参照管理部は、同一 J a v a 仮想マシン内のオブジェクトに対するリモート参照の作成や、他の J a v a 仮想マシン内のオブジェクトを参照するリモート参照を受信し管理するためのものである。

【0023】

ここで、あるサービスを提供するサーバプログラムが J a v a 仮想マシン S 内で作成され、これに、J a v a 仮想マシン C 内のクライアントプログラムがアクセスしたとする。

【0024】

まず、移動（ネットワークアドレスの変更）がない場合について説明する。

【0025】

図 2 は、サーバプログラムが稼働する仮想マシン S の構成を示したもので、図 3 に示すフローチャートを参照して図 2 の仮想マシン S の機能と動作について説明する。

【0026】

図 2 において、仮想マシン S は、主に、サーバオブジェクト 2 と登録管理部 3 とリモート参照管理部 4 とネットワークアドレス獲得部 5 とから構成される。

【0027】

J a v a によるサーバプログラムは、一般に、複数の j a v a オブジェクトで構成されるが、その中に R M I による遠隔メソッド呼び出しを受け付けるオブジェクトが含まれる。このオブジェクトは、「j a v a . r m i . s e r v e r . R e m o t e S e r v e r」クラスを継承したクラスのオブジェクトである。このオブジェクトを以下サーバオブジェクト 2 と呼ぶ。

【0028】

リモート参照管理部 4 は、主に、リモート参照更新部 4 a とリモート参照格納部 4 b とから構成され、リモート参照更新部 4 a は、サーバオブジェクト 2 への

リモート参照の作成、更新を行い、作成、更新されたリモート参照をリモート参照格納部 4 b に格納する。また、リモート参照管理部 4 は、他の J a v a 仮想マシン内のオブジェクトを参照するためのリモート参照を受信し、管理する。

## 【 0 0 2 9 】

登録管理部 3 は、リモート参照格納部 4 b に格納されたサーバオブジェクト 2 へのリモート参照をディレクトリサービスプログラム D に登録するためのものである。

## 【 0 0 3 0 】

ネットワークアドレス獲得部 5 は、サーバオブジェクト 2 が構築、すなわち、起動された（クラスがインスタンス化された）ときに、計算機 1 0 2 のオペレーティングシステム（例えば、Windows（マイクロソフト社登録商標）、Linux 等で、以下、簡単に OS と呼ぶ）から、そのときの計算機 1 0 2 のネットワークアドレスを獲得する。例えば、「java.net.InetAddress」クラスが提供する「getLocalHost()」メソッドがその一例である。Sun Microsystems 社が提供する JDK（Java Development Kit）1. 1 および JDK 1. 2 で用意されているものは、ネットワークアドレスを一度獲得すると、Java 仮想マシンが終了するまで、その獲得したネットワークアドレスを保持し続けるようになっている。

## 【 0 0 3 1 】

リモート参照更新部 4 a は、ネットワークアドレス獲得部 5 で獲得されたネットワークアドレスをリモート参照に書き込む。

## 【 0 0 3 2 】

さて、計算機 1 0 2 に電源が投入されて、Java 仮想マシンによりサーバオブジェクト 2 が構築されると（クラスがインスタンス化されると）（図 3 のテップ S 1）、その際、リモート参照管理部 4 は、ネットワークアドレス獲得部 5 からネットワークアドレスを取り出して、このサーバオブジェクト 2 へのリモート参照を作成し、リモート参照格納部 4 b に格納する（ステップ S 2）。

## 【 0 0 3 3 】

次に、登録管理部 3 が、このサーバオブジェクト 2 へのリモート参照をディレ

クトリサービスプログラムDに登録する（ステップS3）。

【0034】

一方、クライアントプログラムは複数のJavaオブジェクトから構成されており、これを以下、クライアントオブジェクトと呼ぶ。クライアントプログラムは、所望のサービスを提供してくれるサーバプログラムを検索するために、ディレクトリサービスプログラムに接続して、サービス名、サービス属性などをキーとして所望のサーバプログラムを検索する（図4のステップS11）。そして、ディレクトリサービスプログラムに登録されている所望のサーバプログラムへのリモート参照を獲得すると（ステップS12）、リモート参照に記述されたネットワークアドレス等を用いて、当該サーバプログラムに接続し（ステップS13）、サービスの提供を受ける（ステップS14）。

【0035】

以上図3のステップS1～ステップS3までのサーバプログラムの動作は、従来同様である。

【0036】

次に、本発明の実施形態に係るサーバプログラムの動作について、上記の移動（ネットワークアドレスの変更）がない場合と異なる部分について説明する。

【0037】

ここで、サーバプログラムが稼働している計算機102のネットワークアドレスが変更された場合について考える。これは、たとえば携帯型のノートパソコンの様な移動可能な計算機をユーザが移動しながら、あるいはサスペンドの状態を持ち運び、別のIPネットワーク（以下、簡単にネットワークと呼ぶ）に接続した場合に生じる。また、移動できないようなデスクトップ型の計算機等であっても、ダイヤルアップで接続されている場合、電話回線を一旦切断した後に再接続した際に異なるネットワークアドレスが割り当てられる事がある。

【0038】

このような計算機102のネットワークアドレスの変更が、サーバプログラム稼働中に生じたとする（図3のステップS4）。この場合、IPネットワーク移動後もサーバオブジェクト2が再構築されることがないので、リモート参照管理

部 4 内に保持されているサーバオブジェクト 2 への接続のためにリモート参照は、サーバオブジェクト 2 が構築された時点でのネットワークアドレスを保持するため、ネットワークアドレスの変更に伴って、リモート参照の書き換えが必要となる。

#### 【 0 0 3 9 】

また、ネットワークアドレス獲得部 5 は、前述したように、ネットワークアドレスを一度獲得すると、その値を保持し、ネットワークアドレス変更後もその保持された値をリモート参照管理部 4 に報告してしまう。このため、ネットワークアドレス獲得部 5 は、その保持した値とネットワークアドレスを削除し、ネットワークアドレス獲得部 5 が呼ばれた場合にネットワークアドレスを再獲得することが必要である。

#### 【 0 0 4 0 】

すなわち、サーバプログラムが稼働する計算機 1 0 2 のネットワークアドレスが変更されたときに、速やかにリモート参照が更新されるよう、計算機 1 0 2 のネットワークアドレスが更新されたことを検知するメカニズムが必要となる。このメカニズムは、ネットワークアドレスを管理する OS が具備してもよいし、図 6 に示すように、ネットワークアドレスをモニタするプログラム（ネットワークアドレスモニタ 6）が、Java 仮想マシン S 外部に常駐していてもよい。これらの場合には、Java 仮想マシン S 外部からリモート参照管理部 4（リモート参照更新部 4 b）へ、ネットワークアドレスの変更を通知する手段があれば良い。この手段には、ソケットなどの通常のプロセス間通信を用いても良い。または、図 7 に示すように、ネットワークアドレスモニタ 7 を Java 仮想マシン S 内部に持っても良い。

#### 【 0 0 4 1 】

図 6 のネットワークアドレスモニタ 6 の機能について、図 8（a）を参照して説明する。ネットワークアドレスは、計算機 1 0 2 に搭載されている、DHCP（Dynamic Host Configuration Protocol）1 2 や PPP（Point-to-Point Protocol）1 3 のようなプロトコルを用いて必要に応じて取得され、計算機 1 0 2 の OS 1 1 に渡さ

れ、ここで保持される。ネットワークアドレスモニタ6は、OS11に保持されているネットワークアドレスを定期的に読み出し、前回読み出したネットワークアドレスと比較する。前回読み出しネットワークアドレスと今回読み出したネットワークアドレスとが異なるときは、その旨をリモート参照管理部4（リモート参照更新部4b）へソケットなどの通常のプロセス間通信を用いて通知する。それを受けて、リモート参照更新部4bはネットワークアドレス獲得部5を起動して、新たなネットワークアドレスを獲得する。

#### 【0042】

図7のネットワークアドレスモニタ7の機能について、図8（b）を参照して説明する。ネットワークアドレスは、上記同様、計算機102に搭載されている、DHCP12やPPP13のようなプロトコルにて取得され、計算機102のOS11に保持されている。ネットワークアドレスモニタ7は、定期的にネットワークアドレス獲得部5を呼び出し、ネットワークアドレス獲得部5で獲得されたネットワークアドレスを前回のものと比較してネットワークアドレスの変更を検知する。ネットワークアドレスの変更を検知した場合には、リモート参照更新部4bのメソッド呼び出しをして、新たなネットワークアドレスを通知する。

#### 【0043】

図3の説明に戻り、ネットワークアドレスの変更が検知され、新たなネットワークアドレスを取得したリモート参照更新部4bは、リモート参照格納部4aに格納されているリモート参照中のネットワークアドレスを書き換える（ステップS5）。

#### 【0044】

リモート参照格納部4bは、複数のリモート参照を管理しており、大別すると同一Java仮想マシン内のオブジェクトを参照するものと、他のJava仮想マシン内のオブジェクトを指すものがある。ネットワークアドレスの変更に伴って書き換えが必要となるリモート参照は、同一Java仮想マシン内のオブジェクトを指すものだけである。

#### 【0045】

ここで、図5に示すフローチャートを参照して、図3のステップS5のリモ-

ト参照更新部 4 b のリモート参照更新処理動作について説明する。

【 0 0 4 6 】

リモート参照更新部 4 a は、リモート参照格納部 4 a で管理されているリモート参照の中から、同一 J a v a 仮想マシン S 内のもので、ネットワークアドレス更新処理を行っていない、未処理のもの選択し（ステップ S 2 1 ～ステップ S 2 2）、その選択したリモート参照中に計算機 1 0 2 が移動する前の変更前のネットワークアドレスが含まれているときは（ステップ S 2 3）、それを移動後の新しいネットワークアドレスに書き換える（ステップ S 2 4）。

【 0 0 4 7 】

なお、ネットワークアドレスの変更を検知する方法としては、上記のような定期的にモニタするのではなく、移動後最初にネットワークアクセスを行う時点でモニタして検出するのも良い。リモート参照は、計算機 1 0 2 の J a v a 仮想マシン S のプログラムが他の計算機の J a v a 仮想マシンと通信を行う時点で更新されていれば良いので、例えば、図 6 に示した構成の場合、計算機 1 0 2 が J a v a 仮想マシン S 外のプログラムが受信したパケットに応答しようとしたときに、ネットワークアドレスモニタ 6 がネットワークアドレスのチェックを行い、変更されていれば、ネットワークアドレスモニタ 6 がリモート参照更新部 4 のメソッド呼び出しをすれば良い。

【 0 0 4 8 】

以上、ネットワークアドレスの変更に伴うリモート参照中のネットワークアドレスを書き換える場合を説明したが、実際、ネットワークアドレスの変更に伴い、書き換えるべきものは、これだけではない。

【 0 0 4 9 】

まず、図 9 を参照して、J a v a の R M I を用いた通信の概略を説明する。

【 0 0 5 0 】

J a v a の R M I では、計算機 1 0 3 のクライアントプログラムが計算機 1 0 2 のサーバオブジェクト 2 が提供するリモートメソッドを呼ぶ際に、当該サーバオブジェクト 2 用のスタブクラスが必要となる。また、サーバプログラムには、このスタブクラスに対応するスケルトンクラスを有する。クライアントプログラ

ムが実行されている計算機 1 0 3 上に、サーバオブジェクト 2 用のスタブクラスが無い場合、サーバオブジェクト 2 のスタブオブジェクトに含まれるコードベースに指定された場所から、スタブクラスを記述したファイル（スタブファイル）をダウンロードする。スタブオブジェクトには、コードベースとリモート参照を含んでいる。

#### 【 0 0 5 1 】

スタブファイルの保存場所としては、サーバプログラムが稼働する計算機 1 0 2 内にある場合と、異なる計算機にある場合がある。前述したように、計算機 1 0 2 の移動に伴ってネットワークアドレスが変更するような場合に影響を受けるのは、明らかに前者の場合である。

#### 【 0 0 5 2 】

計算機 1 0 3 上のクライアントプログラムにスタブファイルを提供する場合、通常 HTTP (Hyper Text Transfer Protocol) が利用される。そこで、本実施形態では、図 1 0 に示すように、サーバプログラムが稼働する計算機 1 0 2 上で、HTTP にてデータファイルの送受信を行う HTTP サーバ 1 0 4 が並行して稼働する。

#### 【 0 0 5 3 】

図 9 において、計算機 1 0 2 上のサーバプログラムが通信相手のクライアントプログラムに提供すべきサーバオブジェクト 2 用のスタブファイルは、HTTP サーバ 1 0 4 に格納する（ステップ S 1 0 1）。また、サーバオブジェクト 2 用のスタブファイルの格納場所を示すコードベースは、スタブオブジェクトをディレクトリサービスプログラムに登録するために送出する際に、スタブオブジェクトに（コードベースが）付与される（ステップ S 1 0 2、図 3 のステップ S 2）。クライアントプログラムは、図 4 に示したように、ディレクトリサービスプログラムから所望のサーバプログラムのスタブオブジェクトを受け取り（ステップ S 1 0 3）、そのスタブオブジェクトに付与されているコードベースを基に、スタブファイルの供給元にアクセスして、スタブファイルをダウンロードする（ステップ S 1 0 4）。その後、クライアントプログラムは、図 4 のステップ S 1 3 以降の動作を行う。

## 【 0 0 5 4 】

さて、サーバオブジェクト 2 用のスタブファイルの格納場所は、計算機 1 0 2 にある。従って、前述のようにして、サーバプログラムが稼働する計算機 1 0 2 のネットワークアドレスが変更された場合、前述同様にリモート参照の書き換えが必要であるが、さらに、スタブファイルを提供する HTTP サーバ 1 0 4 のネットワークアドレスも変更されるため、スタブファイルの供給元を示すコードベースも書き換える必要がある。

## 【 0 0 5 5 】

RMI では、RMI 用クラスローダが定義されている。RMI 用クラスローダは、RMI 用のスタブクラスやスケルトンクラス、RMI のメソッドのパラメータや戻り値として渡されるクラスを管理するものである。具体的には、クラスファイルの在処を、所定の順番に検索、ロードし、合わせてどこからロードしたかを記憶する。そして、オブジェクトを他の J a v a 仮想マシンへ渡す場合、受け手側でクラスがロードできるように在処を示す URL を付与する。この時、スケルトンやスタブなどのクラスに対して付与されるコードベースは RMI クラスローダが管理するローカルコードベースの値がコピーされる（詳細は RMI 仕様書）。このローカルコードベースは、初期値は起動時にプロパティで指定するのが一般的である。

## 【 0 0 5 6 】

図 1 1 に示すように、サーバプログラムが稼働する仮想マシン S は、上記の RMI 用クラスローダ 8 を有する。なお、図 2、図 6 と同一部分には同一符号を付し、以下、異なる部分について説明する。計算機 1 0 2 のネットワークアドレスが変更された場合には、RMI 用クラスローダ 8 が管理しているローカルコードベース 8 a の値が更新されればよい。そこで、ネットワークアドレスの変更を検知した場合には、ローカルコードベース更新部 8 b を呼んで、ローカルコードベース 8 a に含まれる自計算機 1 0 2 を示す部分を、新しいネットワークアドレスに対応したものに更新すればよい。

## 【 0 0 5 7 】

例えば、計算機 1 0 2 のネットワークアドレスが「1 0 0 . 1 0 1 . 1 0 2 .

103」、コードベースが「http://100.101.102.103/codebase/」であったものが、ネットワーク移動により、計算機102のネットワークアドレスが「200.201.202.203」に移動になった場合、コードベースも「http://200.201.202.203/codebase/」に更新されれば良い。

#### 【0058】

ネットワークアドレス変更の検知は、図8に示したように、Java仮想マシンS外部で行う方法（図8（a）参照）とJava仮想マシンS内部で行う方法（図8（b））とがあるが、図11では、Java仮想マシンS外部にネットワークアドレスモニタ6がある場合の構成例を示している。すなわち、前述したように、ネットワークアドレスモニタ6がネットワークアドレスの変更を検知すると、その旨をリモート参照管理部4（リモート参照更新部4b）、RMI用クラスローダ8（コードベース更新部8b）へソケットなどの通常のプロセス間通信を用いて通知する。それを受けて、リモート参照更新部4b、コードベース更新部8bは、ネットワークアドレス獲得部5を起動して、新たなネットワークアドレスを獲得する。コードベース更新部8bは、獲得した新たなネットワークアドレスを用いて、ローカルコードベース8aを書き換える。

#### 【0059】

図3のステップS5では、リモート参照更新部4bは、リモート参照中のネットワークアドレスを新たなネットワークアドレスに書き換えるとともに、コードベース更新部8bでローカルコードベース8aを更新する。

#### 【0060】

以上のようにして、リモート参照更新部4bにより更新されたりリモート参照は、計算機102の移動先のIPネットワーク上に存在するディレクトリサービスプログラムに登録される（図3のステップS3）。なお、このときのディレクトリサービスプログラムは、計算機101上で稼働するものであるとは限らない。

#### 【0061】

このように、計算機102の移動に伴うIPアドレスの変更に応じて、リモート参照更新部4aがリモート参照中のネットワークアドレスを書き換えて、計算

機 1 0 2 の移動した先においても、スタブオブジェクトにその新たなりモート参照と新たなローカルコードベースを付与して、ディレクトリサービスプログラムに登録することにより、計算機 1 0 2 上のサーバプログラムは、新たなクライアントとの間の通信が可能となる。

#### 【 0 0 6 2 】

以上説明したように、上記実施形態によれば、サーバプログラムを実行する J a v a 仮想マシン S を少なくとも 1 つ有する計算機 1 0 2 が、ネットワークアドレスモニタ 6 あるいは 7 で自装置の I P アドレスの変更を検知したとき、サーバプログラムへのリモート参照中のネットワークアドレスとコードベースの更新を行い、この更新されたりモート参照とローカルコードベースを含むスタブオブジェクトをディレクトリサービスプログラムへ登録し直すことで、計算機 1 0 2 の I P アドレスの変更後も、このサーバプログラムへのクライアントからのアクセスが可能になる。

#### 【 0 0 6 3 】

なお、上記実施形態では、スタブファイルを例にとり説明したが、この場合に限らず、例えば、サーバからクライアントへのサービス提供の際、クライアントがサーバに対し渡す引数に、例えばその解釈のために所定のクラスファイルが必要なときに、そのクラスファイルの所在を記述したデータを当該引数と伴に渡すことになるが、クライアントプログラムの稼働する計算機の移動に伴い I P アドレスが変更した場合、当該クラスファイルが同じ計算機内に存在するときには、前述同様にして、このクラスファイルの所在を記述したデータも変更して、サーバプログラムへ送り直せばよい。

#### 【 0 0 6 4 】

また、サーバがクライアントから受け取った引数に対し行った処理結果に、例えばその解釈のために所定のクラスファイルが必要なときも、そのクラスファイルの所在を記述したデータを当該処理結果と伴に渡すことになるが、サーバプログラムの稼働する計算機の移動に伴い I P アドレスが変更した場合、当該クラスファイルが同じ計算機内に存在するときには、前述同様にして、このクラスファイルの所在を記述したデータも変更して、クライアントプログラムへ送り直せば

よい。

【 0 0 6 5 】

なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。さらに、上記実施形態には種々の段階の発明は含まれており、開示される複数の構成要件における適宜な組み合わせにより、種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題（の少なくとも1つ）が解決でき、発明の効果の欄で述べられている効果（の少なくとも1つ）が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【 0 0 6 6 】

【発明の効果】

以上説明したように、本発明によれば、プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも1つ有する端末のネットワークアドレスが、例えば上記プログラム実行中に不意に変更されても、そのネットワークアドレスの変更を上記実行環境上に反映して、後に、再び他の端末との通信を可能にする。

【図面の簡単な説明】

【図 1】

本実施形態に係る分散情報処理システムの構成例を示す図。

【図 2】

従来のサーバプログラムが稼働する仮想マシンの構成を示した図。

【図 3】

サーバプログラムの処理動作を説明するためのフローチャート。

【図 4】

クライアントプログラムの処理動作を説明するためのフローチャート。

【図 5】

サーバプログラムのリモート参照更新処理動作を説明するためのフローチャート。

【図 6】

本実施形態に係るサーバプログラムが稼働する仮想マシンおよびそれを有する計算機の構成を示した図。

【図 7】

本実施形態に係るサーバプログラムが稼働する仮想マシンおよびそれを有する計算機の他の構成を示した図。

【図 8】

ネットワークアドレス変更の検知方法を説明するための図。

【図 9】

J a v a の R M I を用いた通信の概略を説明するための図。

【図 1 0】

本実施形態に係る分散情報処理システムの他の構成例を示す図で、サーバプログラムが実行される計算機上に H T T P サーバを有する場合を示している。

【図 1 1】

本実施形態に係るサーバプログラムが稼働する仮想マシンおよびそれを有する計算機のさらに他の構成を示した図。

【符号の説明】

S、C、D… J a v a 仮想マシン

2…サーバオブジェクト

3…登録管理部

4…リモート参照管理部

4 a…リモート参照格納部

4 b…リモート参照更新部

5…ネットワークアドレス獲得部

6、7…ネットワークアドレスモニタ

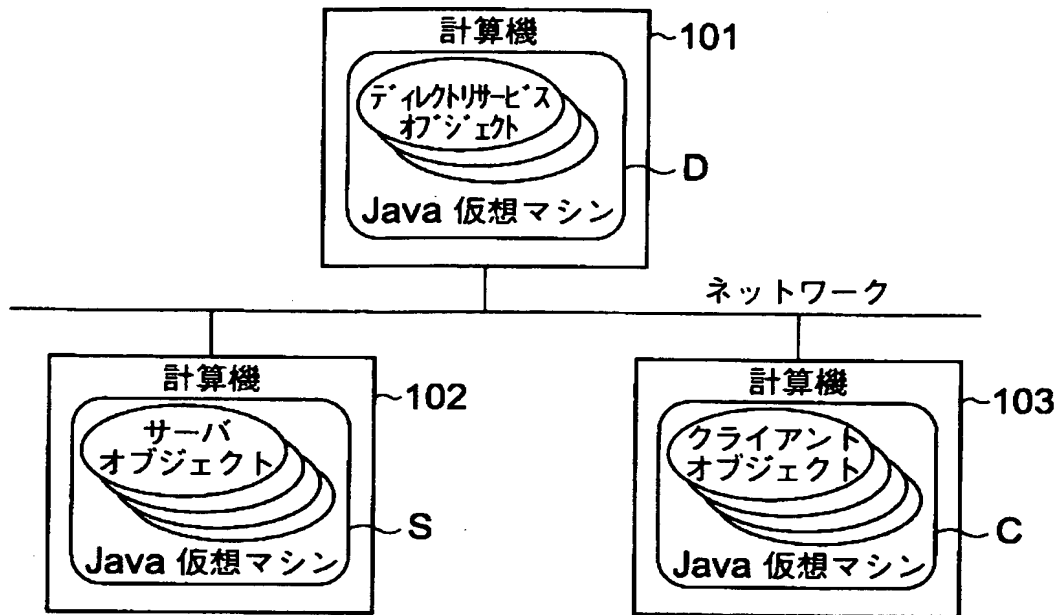
1 0 1、1 0 2、1 0 3…計算機

1 0 4…H T T P サーバ

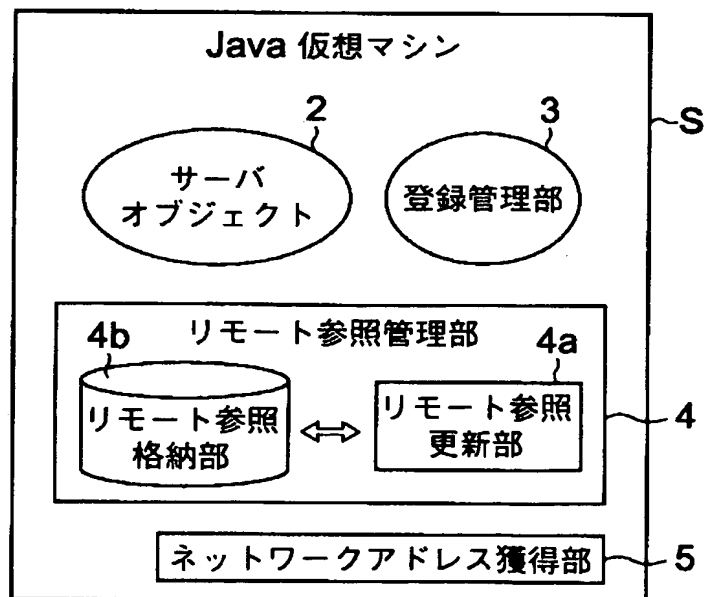
【書類名】

図面

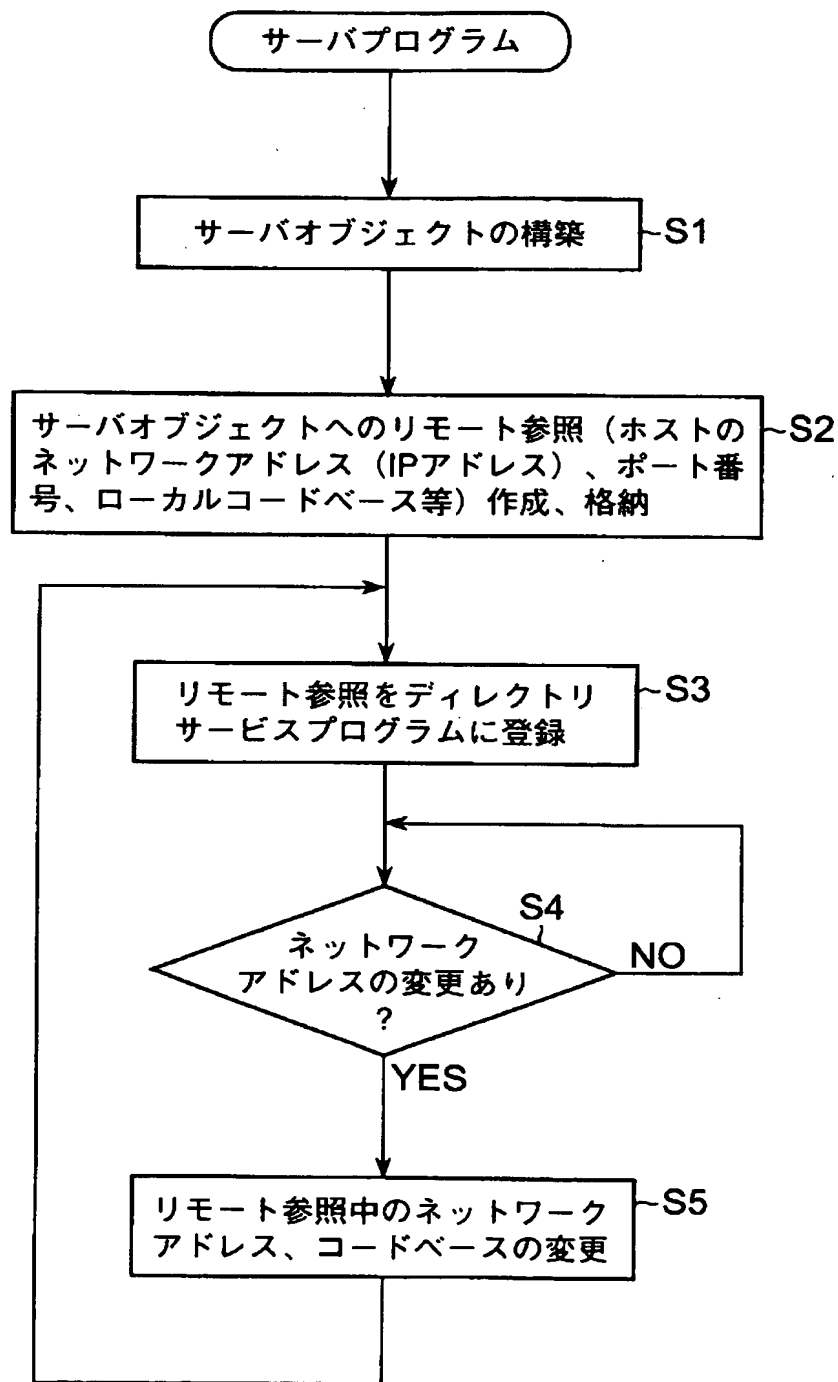
【図 1】



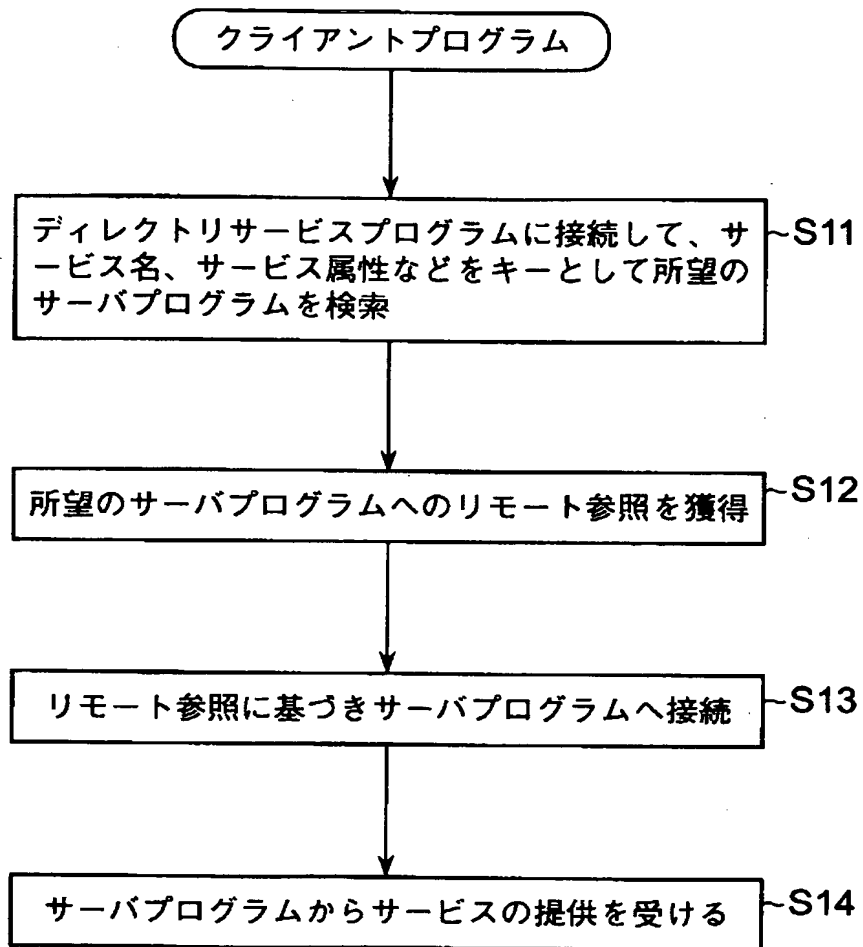
【図 2】



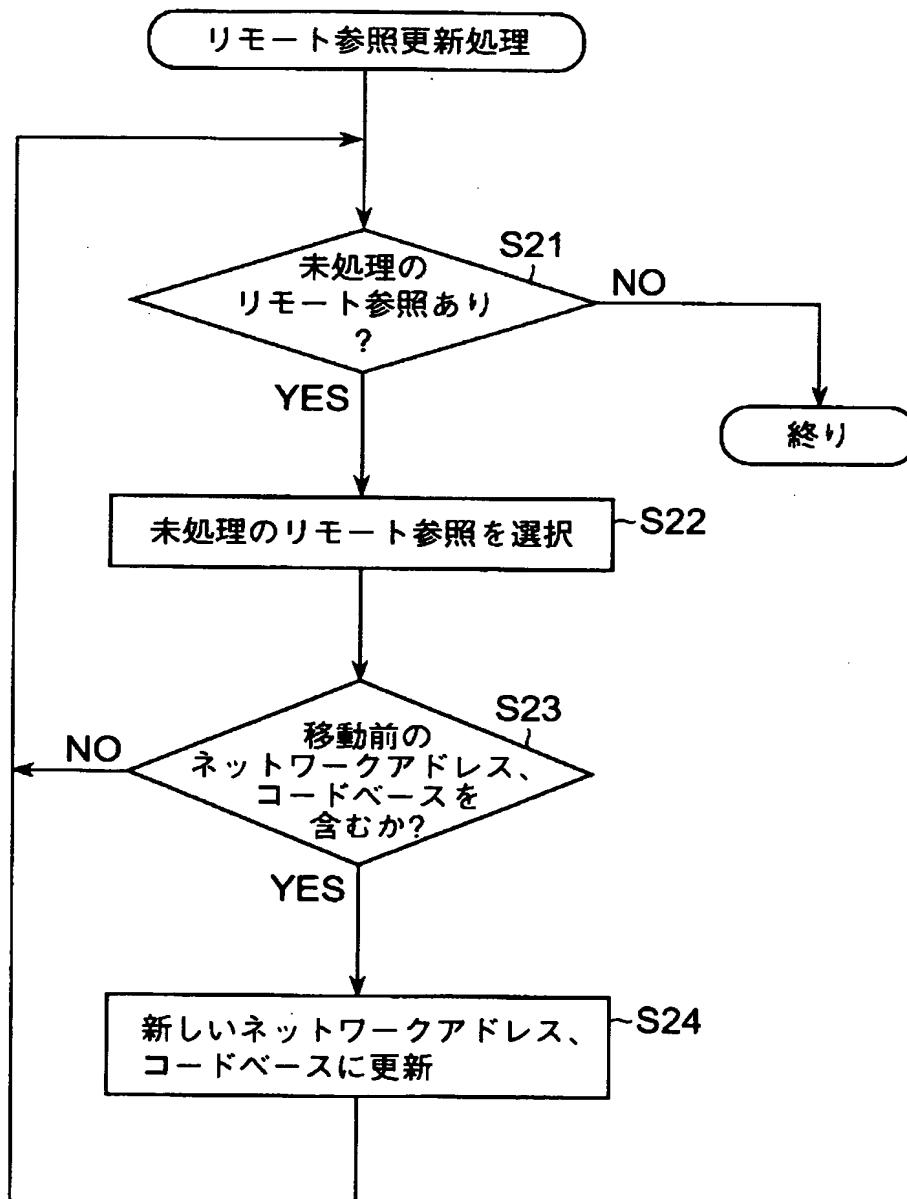
【図3】



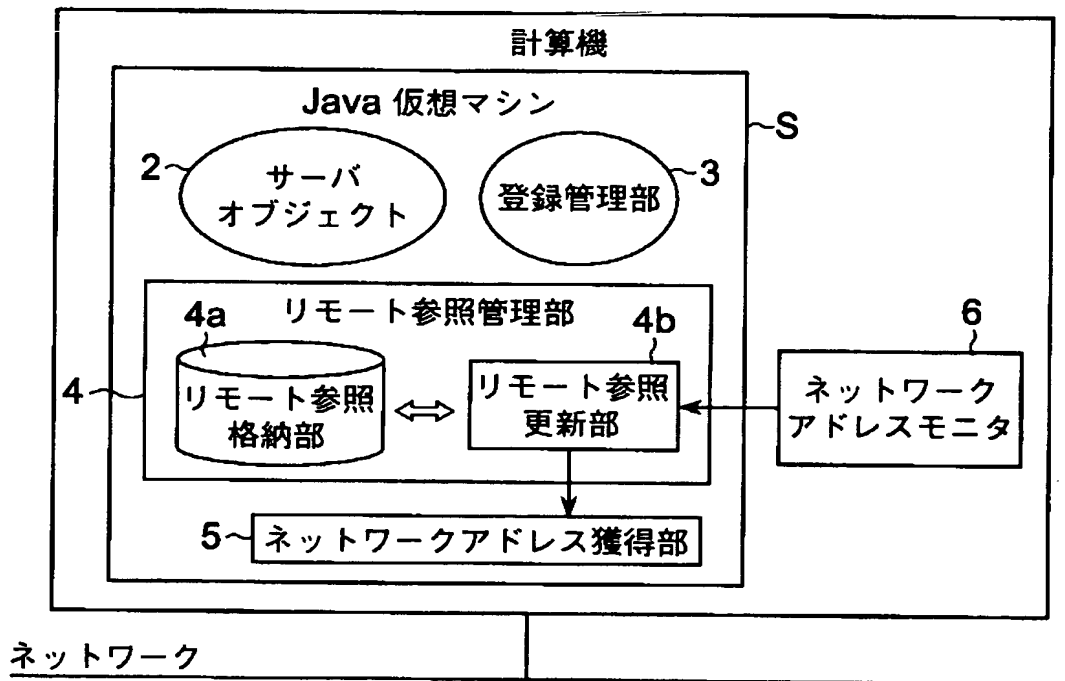
【図 4】



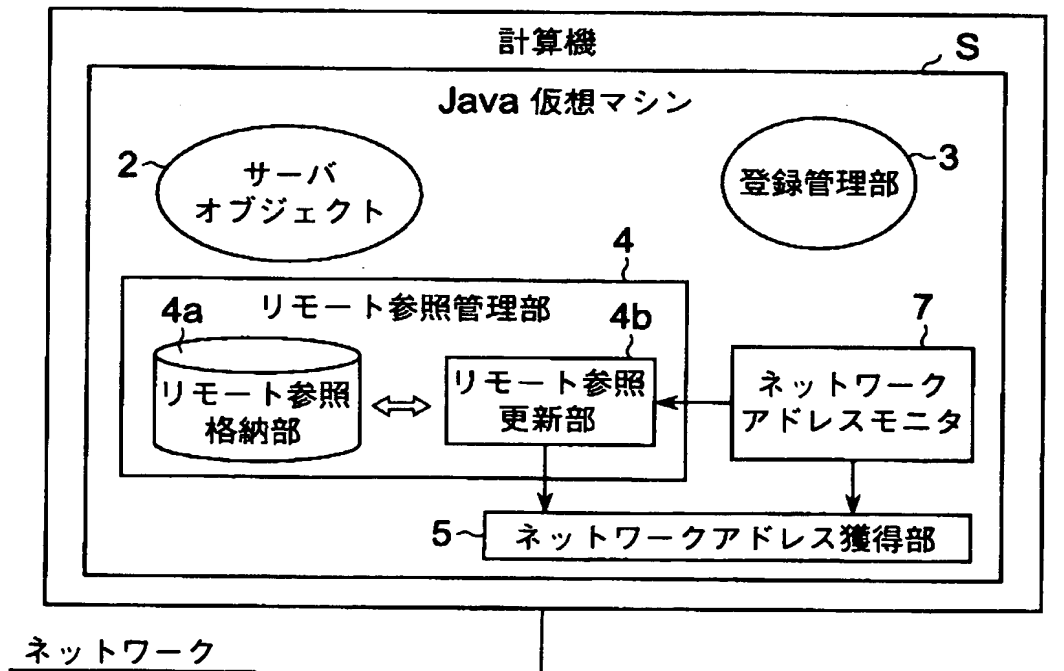
【図 5】



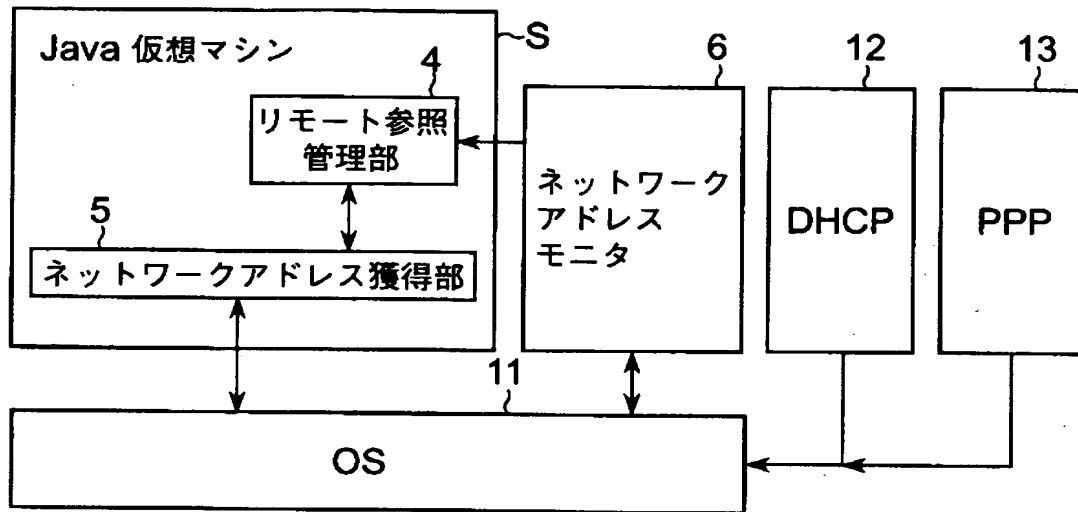
【図 6】



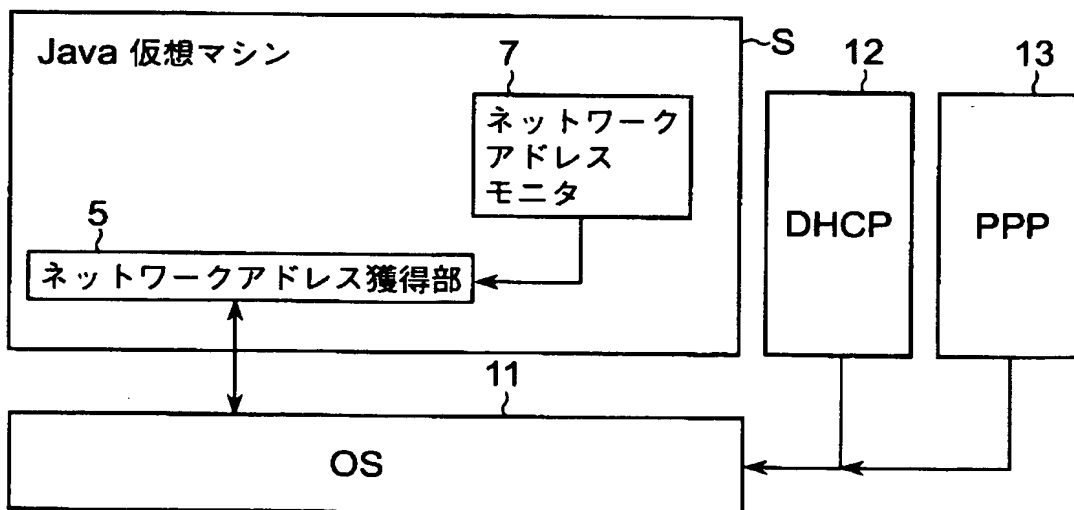
【図 7】



【図 8】

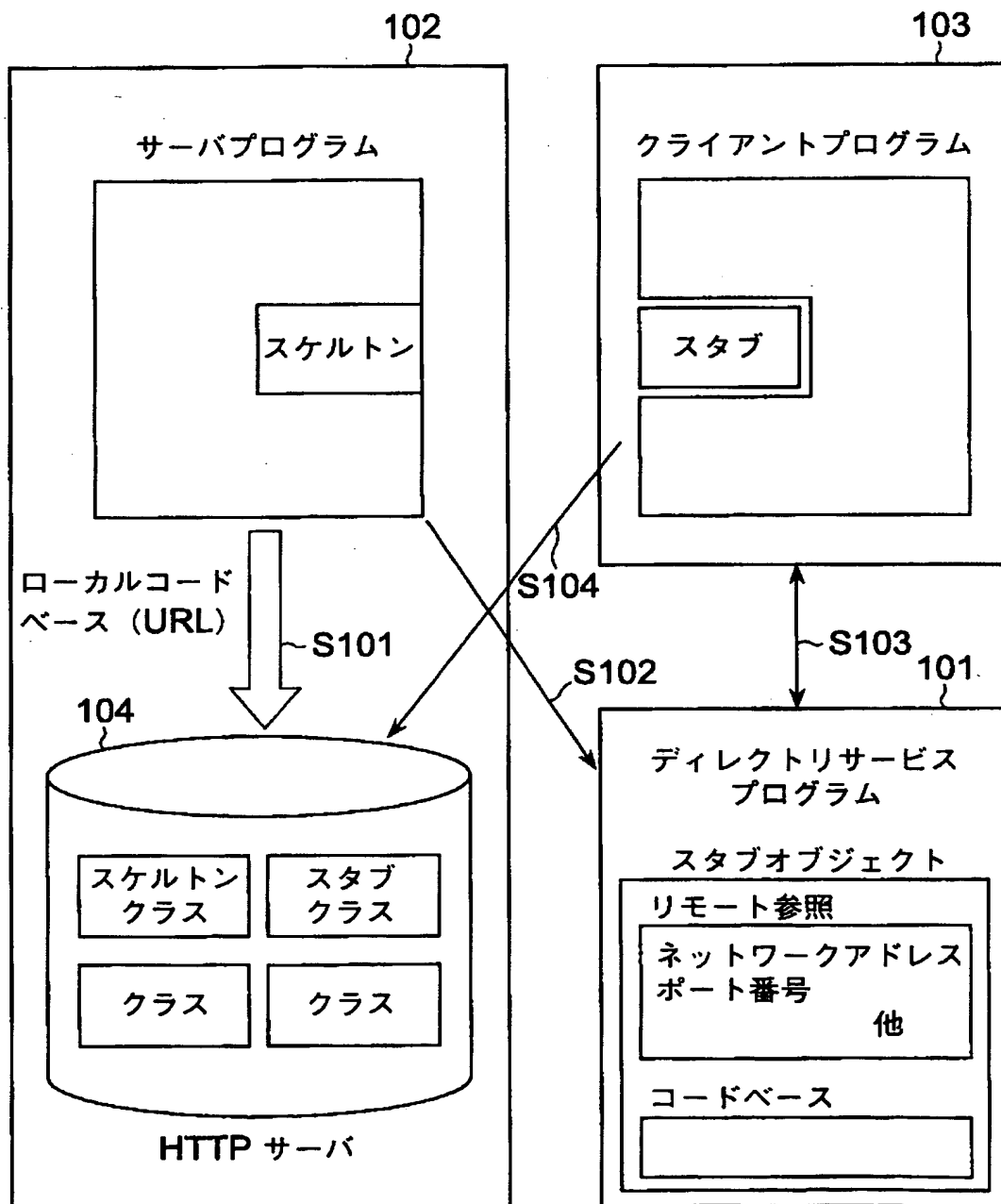


(a)

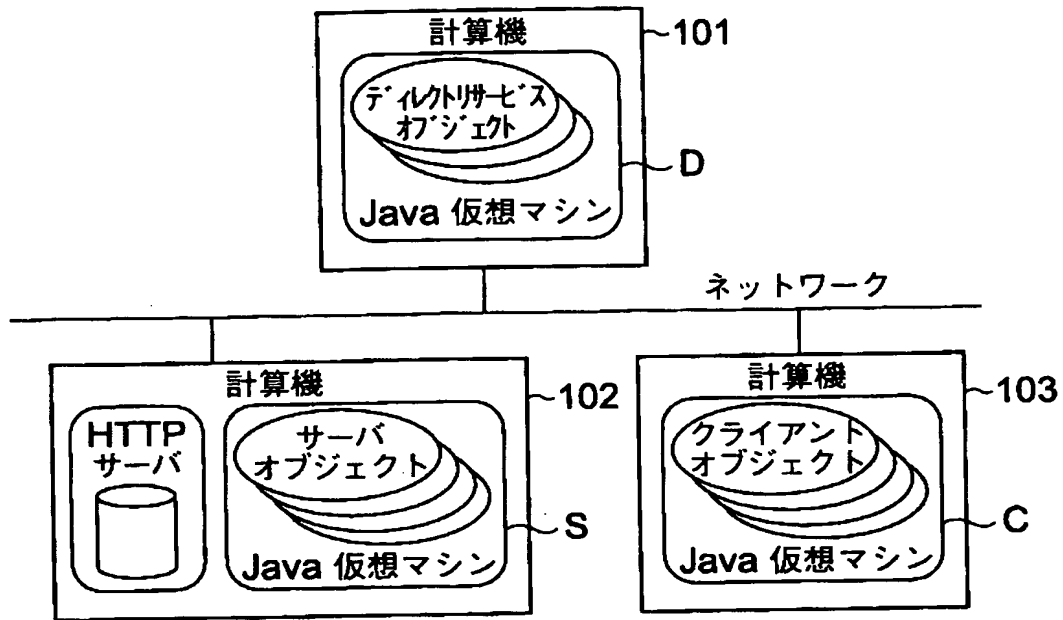


(b)

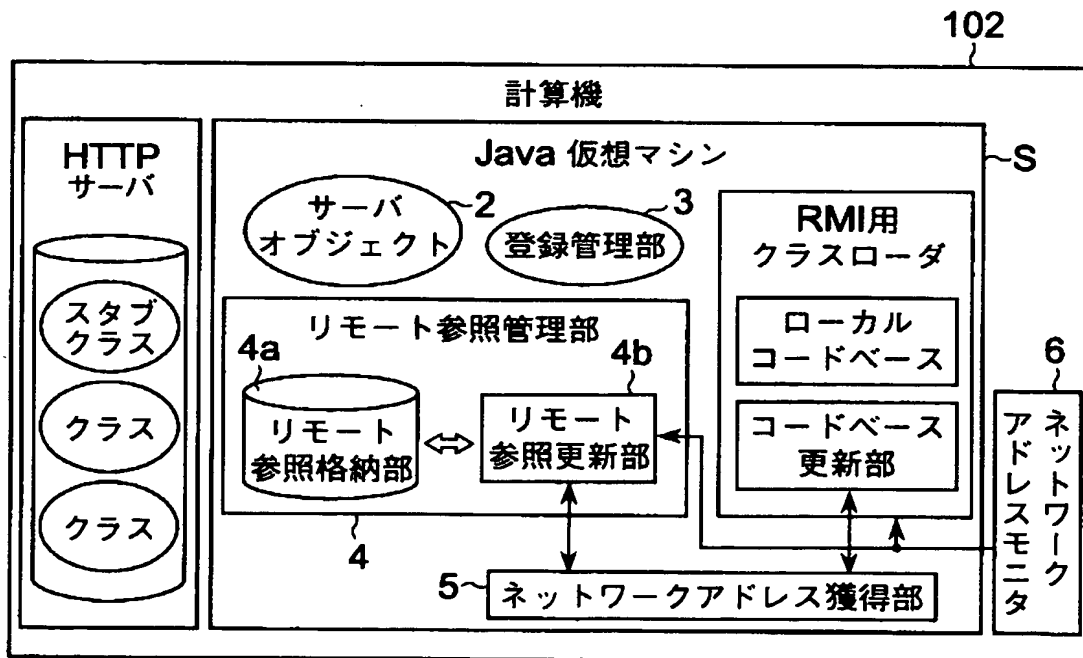
【図 9】



【図10】



【図11】



【書類名】                      要約書

【要約】

【課題】 プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも1つ有する端末のネットワークアドレスが、例えば上記プログラム実行中に不意に変更されても、そのネットワークアドレスの変更を上記実行環境上に反映して、後に、再び他の端末との通信を可能にする通信方法および情報処理装置を提供する。

【解決手段】 プラットフォーム独立の機械語で実行されるオブジェクト指向言語で記述されたプログラムを実行する実行環境を少なくとも1つ有する複数の端末があり、これらのうちの特定の端末が、自装置のネットワークアドレスの変更を検知したとき、他の端末が前記特定の端末へのアクセスを可能にするため、変更後の新たなネットワークアドレスを前記他の端末が参照可能なように提供する。

【選択図】                      図 6

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日	1990年 8月22日
[変更理由]	新規登録
住 所	神奈川県川崎市幸区堀川町72番地
氏 名	株式会社東芝